

Flexible Data Input Layer Architecture (FDILA) for Quick-Response Decision Making Tools in Volatile Manufacturing Systems

Siamak Tavakoli, Ali Mousavi, Alexander Komashie

Systems Engineering Research Group
School of Engineering and Design, Brunel University
Uxbridge, United Kingdom
Siamak.Tavakoli@Brunel.ac.uk

Abstract—This paper proposes the foundation for a flexible data input management system as a vital part of a generic solution for quick-response decision making. Lack of a comprehensive data input layer between data acquisition and processing systems has been realized and thought of. The proposed FDILA is applicable to a wide variety of volatile manufacturing environments. It provides a generic platform that enables systems designers to define any number of data entry points and types regardless of their make and specifications in a standard fashion. This is achieved by providing a variable definition layer immediately on top of the data acquisition layer and before data pre-processing layer. For proof of concept, National Instruments' Labview data acquisition software is used to simulate a typical shop floor data acquisition system. The extracted data can then be fed into a data mining module that builds cost modeling functions involving the plant's Key Performance Factors

Data Input Layer, Real-Time Data, Decision Support, Volatile Manufacturing Industry, Production Management

I. INTRODUCTION

Reference [1] proposed a solution for information integration in multi-sensor systems. They however focused on the architecture aspects of distributed sensor networks with regarding to sensor data clock synchronization, communication costs, and fault tolerance. Data Integration was also addressed by [2] who proposed a query processing algorithm as a semantic solution in wireless smart sensors. In both solutions processing of entered sensor data necessitates inefficient method of depositing the data into a database accompanied by all the acquired data attributes which significantly increases computational cost.

Other researchers have focused on the security of data over sensor networks [12], efficiency of data transmission [13], dependability of sensor networks [14], independence of sensor data [15] etc amongst others. Our focus in this paper on the other hand is to provide a flexible environment for filtering data from the numerous sensor points in a way that will facilitate quick decision making.

The need for a robust tool that helps production managers make quick decisions in fast moving consumer goods (FMCG) industry has been discussed in [3] which introduces a platform

that deals with food manufacturing environments. Following the same research pathway, a generic framework is introduced here that intends to be applicable to wider manufacturing environments. In addition to a more robust input layer management system the proposed generic framework aims at providing comprehensive resource-based cost models.

The raw data extracted directly from sensor output is in the form of an analogue or digital voltage signal. This signals may directly be connected to programmable logic controllers (PLC) or computers (equipped with specialist drivers and software) where they are converted into system specific parameters such as temperature, counter, etc. In the case of smart sensors this data conversion may be implemented locally by attached micro-controllers. In both cases there needs to be a specified communication management standard that manages the data handling procedure to and from the control unit. The data handling and management process is called the Data Interface Unit (DIU). Beyond DIU there is no further reference to scanning and reading sensor data in the Plant Information System Apparatus (PISA). The input data from DIU is then converted into the type and format suitable for the system performance analysis functions. For example, two or more data entities may be combined to build a new data variable. Combined together, the Data Fusion Unit (DFU) and the DIU form the Data Integration Module (DIM).

Figure 1 shows how the DIM integrates the network of

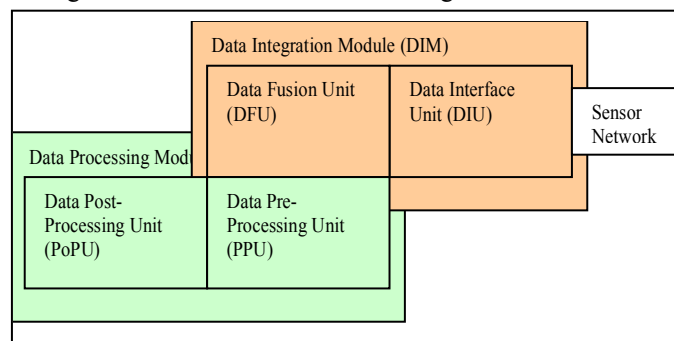


Figure 1. Data Analysis System Overview

plant sensors with the data processing module (DPM).

In the proposed architecture the DPM is used to deposit input data coming from the DIM into a database. The DPM consists of two independent units, the data Pre-Processing Unit (PPU) and the data Post-Processing Unit (PoPU). The PPU is designed to implement the data mining and statistical analysis (i.e. curve fitting) methods. The resultant information from the PPU will be used for effective condition monitoring and process control undertaken in the data Post-Processing Unit (PoPU).

Technically the DIM and the DPM are designed as two autonomous (encapsulated) blocks, but there exists a logical cause effect relationship between them. For example, according to the decision made by an Input Variable Monitoring and Selection task in Data Processing stage, scanning rate of some data entities may change. This implies that Data Fusion may receive influence from Data Pre-Processor on how to fuse input data so that they fit the particular analysis solution.

Reference [4] developed an integrated system for manufacturing data collection and statistical process control using “off the shelf” software packages. They however, did not consider any data fusion layer with a degree of flexibility. Only data acquisition interface was able to be reconfigured upon new process layout. Their system was appropriately dedicated to the automation of an etch area that includes polysilicon, oxide, and metal etching.

In the following sections the main architecture of a practicable software solution to address the need for a universal field sensor data fusion tool is discussed. The proposed FDILA may lead to better standardisation and reduce computational costs. For proof of concept at this stage of the research and development, a simulation project is designed to assess the advantages and the shortcomings of the proposed architecture.

At the earlier stage of this research we suggested an input variable monitoring system for controlling the vast number of inputs to the cost model in order to optimize computational overhead of the whole system [5]. Implementation of such Data Fusion Unit will also help us with proving this comprehensiveness, since it provides an environment in which, several different scenarios of sensor data collection can be implemented.

II. DATA INTEGRATION

Computational overhead of sensor data pre processing and post processing depends on the architecture of sensor network as well as how the sensor data is integrated [1]. This becomes highly important when the population and the variety of sensors are larger. An efficient data integration strategy would become a great concern where there is no option of reconfiguring the designed sensor network.

While in a modern manufacturing system the main processes are locally controlled, still many data points are networked to have their data acquired and gathered for a further central process. A typical shop floor data collection system may be designed to for example gather data about

process yield, machine performance, operation times, order status, inventory, product traceability, product/process quality, and personnel [6]. The potential problems and their causes of systematic input data collection in shop floors are discussed in [7]. Their problems and their probable causes can be erroneous problem definition, lack of clear objectives, system complexity, unwarranted systems, poor data access, difficulty in identifying available data sources, and limited data handling capability. Reference [7] developed a reference data model that links parts and resources in a production flow. However, the proposed system falls short of explaining the methods that relate sensory input data to their proposed input data model.

The data integration process (data fusion) involves the combining of incoming sensor data and building new data variables. The proposed generic solution presented here consists of a virtual environment that enables the user to build data variables in any configuration based on system requirements. This is consistent with the outline presented in [2]. Thus, there can be an initial interaction session between the user and the system for data variable configuration with regard to input entities. Such interaction may be more efficient if it requires minimal expertise in connecting data acquisition (DAQ) devices when defining system variables. And similarly, it would require minimal expertise from people who install DAQ devices of which data points connect to system variables. Layered architecture is a well known approach to gathering and preparing data for different types of volatile systems. In Figure 2 DIU layer ensures data is prepared for fusion. Essentially, DIU layer acquires data from each sensor

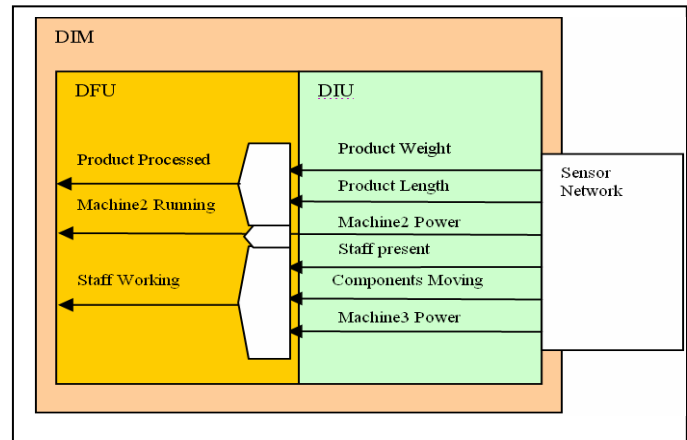


Figure 2. Data Integration main functional stages

node in a regular fashion with scanning frequency set and controlled by DFU layer. System variable definition requires the fusion of sensor data that stem from signal conditioning and requires a modular environment. This guaranties fast maintenance and reconfiguration when a system's processes and components are modified as time passes.

A. DIU and Data Acquisition (DAQ)

The DAQ equipment responsible for sensor-based data

acquisition may use more than one method to collect, buffer and communicate with the main computer. In a generic solution, as long as the real-time data is available, data type and the DAQ firmware as well as the communication channels should not matter. However, to maintain the quality of measurement, the accuracy of DAQ systems depends on:

- 1) *Temperature*: The accuracy of DAQ systems rely on device temperature.
- 2) *Quantization error*: There is a direct relationship between the bit resolution of DAQ device with the accuracy of reading the real physical value.
- 3) *Settling time*: Is the time it takes a DAQ device to amplify an input value to the final (expected) level. This can affect readability of values at high scanning rates.
- 4) *Linearity*: Gain and offset errors would need a set of simple calibration tasks to compensate for losses.
- 5) *Non-linearity*: Non-proportional changes of the signal versus changes of the physical phenomenon may need further compensation in the form of hardware and/or software.

On a smart sensor node, the electrical signal converted from the physical event can be pre-processed at the node. This pre-processing may include some signal conditioning tasks and avoid complicated and time consuming functions especially when real-time data is needed. A simple transducer sensor or what is called the dumb sensor on the contrary only sends the converted electrical signal as raw data over the sensor network to the DAQ system. Although the present trend in sensor technology has shifted towards smart sensors [8], there are still many sensor outputs in the manufacturing shop floors that are directly connected to a DAQ device. Nevertheless, both dumb sensor data and smart sensor data may be gathered in one data pool by DIU and would be available in arrays with enough attributes for each data entity to the DFU. Sensor data attributes may include ID, location, timestamp (for smart sensors with their own clock, synchronization used, for example; they may send their time), sensor type, and finally the measurand itself.

B. DFU and Sensor Data Aggregation

Pending on the information structure of a manufacturing system and its customised key performance indicators (KPI) sensor data may not on its own be useful. For a singular KPI there may be a number of data-points received from corresponding sensor. These KPIs could include machine name, resource status, part count, work shift patterns, quality factors, process times, machine down times, resource utilization, and production throughput, [9]. After obtaining the necessary attributes from sensor data in DIU, the DFU may then reads them and combines by system specified functions and conditions. A configuration interface with a compilation engine would produce meaningful information that becomes

available to the PPU A run-time engine interprets the relations and prepares values for the system variables. In the system variable definition stage a user is enabled to define a variable that is meaningful to the data processing module, and also select inputs among a list of available inputs. The user can then continue with selecting the type of transform on any selected input, and also select how to aggregate result of each transform. Figure 3 below illustrates an example of

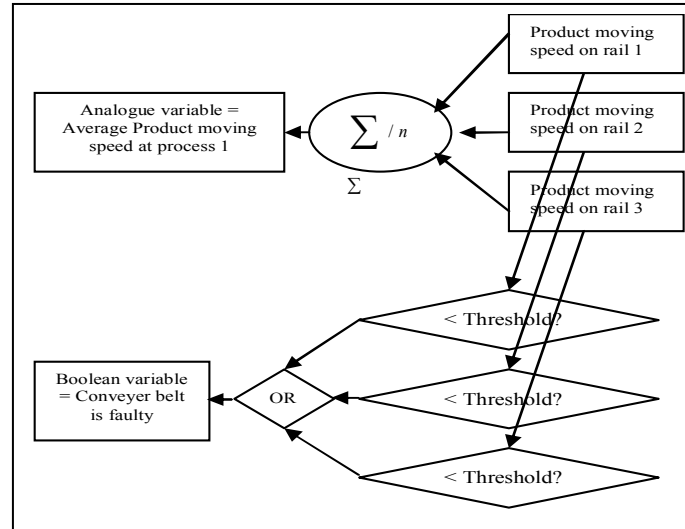


Figure 3. Data Integration main functional stages

aggregating sensor data into variables.

The above data aggregation concept is not unfamiliar to the well established Hierarchical Object-Oriented Design (HOOD) [10]. This structure forms the basis for the Flexible Data Input Layer of the data fusion concept in our generic methodology. One must not mistake this concept with knowledge editing and inference engine of knowledge based and rule based information systems [11], since there is no control strategy and mechanism to be constructed here. Instead, as a low-level or data-level data pre-processing, this concept can be compared to data mining techniques such as feature extraction and combinatorial search. The difference is that historical data is used at later stages.

C. Pros and Cons of Proprietary Data Integration Systems

- Pros;

1) *Rapid development*: They may be used for quick solution developments with ready to use materials and functionality.

- Cons;

1) *Too technical for product management level*: A product manager would not need to configure a data acquisition system at as low level as dealing with ports and pins as expected in existing data integration systems.

2) *Proprietary DAQ firmware:* NI's Labview needs Labview Driver software to be installed as a run-time engine. ML's DAP acquires data from only ML's DAQ hardware. There must be a generic way of introducing sensor data to DAQ software with no need for a particular manufacturer's DAQ firmware.

3) *General purpose:* After all, NI's Labview is a visual programming environment and therefore, introduces more than required functionalities. This complexity lends itself to difficulties to find the proper route between many available options. A more specifically designed environment obviously facilitates quick decision making.

III. IMPLEMENTATION

The objective is to propose a DIM using "off the shelf" software application development environment packages that are independent of proprietary hardware and software. Reference [10] developed a hybrid shop floor control system for food manufacturing. They implemented National Instruments' Labview DAQ software and hardware in order to retrieve sensor data from an ice cream freezer process. Using the current versions of the Labview software, data acquisition from standard PC port connections has become independent of NI's proprietary data acquisition hardware. Additionally, Labview is introduced as a visual programming language for developing input data interface and processing applications. Therefore, Labview is used to simulate data inputs through Virtual Instruments and Signal Generators. This is supposed to satisfy the functionality of our Data Interface Unit. A Data Fusion Unit in Labview software is also developed to simulate the output data which transfers data points from DIU to the DFU.

A. The Test Bed for the Sensor Network

An automated pet food manufacturing environment is selected as a case study for proof of concept.

According to this case, as it is modelled in Figure 4, there are 34 processes that take place at 17 stations. The manufacturing stations in this process include a pre-breaker, two Grinders, six Silos, two Mixers, three Pumps, three Sievers, three Emulsion Hopper, three Tunnels, three Fillers, three Seamers, one FMC, one Stork, two Labellers, three Kisters, two Palletisers, and one Wrapper.

The key data available from the production process include the process time for each machine, product route and transfer times, machinery capacity, the inter-arrival between each order. The acquired information can lead to estimation of throughput time, resource utilization, silo level, and average production rate.

B. Time Factors

Processing time factors and transfer time factors may be possible to work out from traceability of product and load condition of machinery. RFID based proximity detection can help with sensing presence of one particular product at a

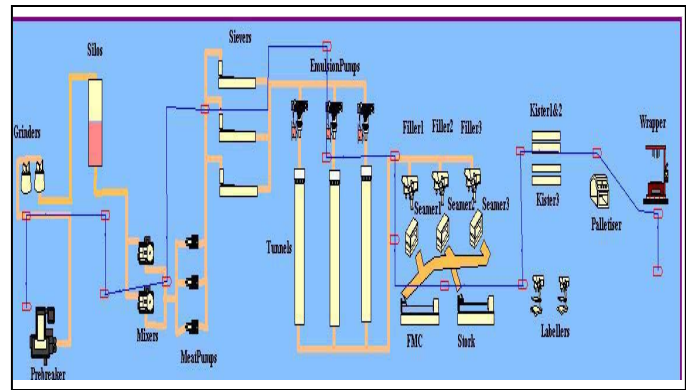


Figure 4. Friskies Nestle Meat Factory Model

particular stage. Light Reflection Detection Sensors (LRDS) may also help with announcing the entry of products into a stage. In this case prior to the meat being minced and the process converting from discrete into a continuous flow, LRDS are used in front of each processing machine or conveyer belt to detect arrival of meat pieces. These sensors are on/off switches with two level of electrical voltage. One transition from on to off or vice-versa instigating the signal required to describe the introduction of one pallet of meat to the production process.

If a machine starts processing a product the work load and / or working condition of the machine may change to some extent, so it can be detected by some appropriate sensor. For example, a movement speed, a rotation speed, a temperature, or pressure may change to a sensible extent. Alternatively, the amount of current sought by a motor can be different compared to when it is not processing. Electrical current sensors may be installed on motor drive circuit of a machine, so that while the machine is processing, change of the current drawn through their motors can be detected. This can generate a multi-level signal, which may be compared against a pre-defined threshold value for deducing if the machine is processing or not.

C. Capacity Detection

The capacities of silos are read from load cells. The detectors all have an analogue or digital output indicating the level of minced meat in the silos. Level of meat in the filled cans may be measured using range detectors. The analogue output may be compared against a threshold value for understanding if the can is filled.

D. Conveyor Speed

The continues flow of processed meat on the conveyer belt after the grinding process may be measured by the speed of conveyer which can be obtained from conversion of rotational speed of driver motors into linear movement of the belt. Rotational speed is provided from shaft encoders by counting the number of generated pulses. Some dimensions and threshold values may be fed to the Data Fusion system as fixed information.

E. Selection of Sensors

Table 1 defines which sensors are selected for our implementation. Light Reflection sensor indicates presence of an object in front of its light beam. Therefore this sensor generates a two level output data type. It is selected for entry and end stage of some machines and conveyer belts. A current sensor is selected for detecting load work of motor driver of some machines. It generates an analogue signal with several voltage levels depending on the electrical current. Such multi-level analogue signal may be digitized by Data Acquisition hardware before entering DIU. Range Detector is selected for level detection of filling meat in silo or filled can. It also generates a multi-level signal. Speed Detectors with multi-level outputs are selected on some conveyer belts. Finally, switches are selected to indicate presence of the object with similar output to LRDS.

F. DIU Development

21 bi-level signals and 11 multi-level digital signals with 8-bit resolution can be acquired using a hardware interface which can read 15 8-bit bytes from several ports of a computer. 21 bits can also be read from three 8-bit ports and the other 11 bytes occupy one 8-bit port each. A Virtual Instrument can be designed and developed in NI's Labview software in order to read bytes from computer ports. In figure 5 partial schematic of designed interface is shown. Attributes are added to serial port reader blocks in order to define port settings such as port address and scan rate. The first 4 bits of the first byte represent values from Prebreaker Entrance,

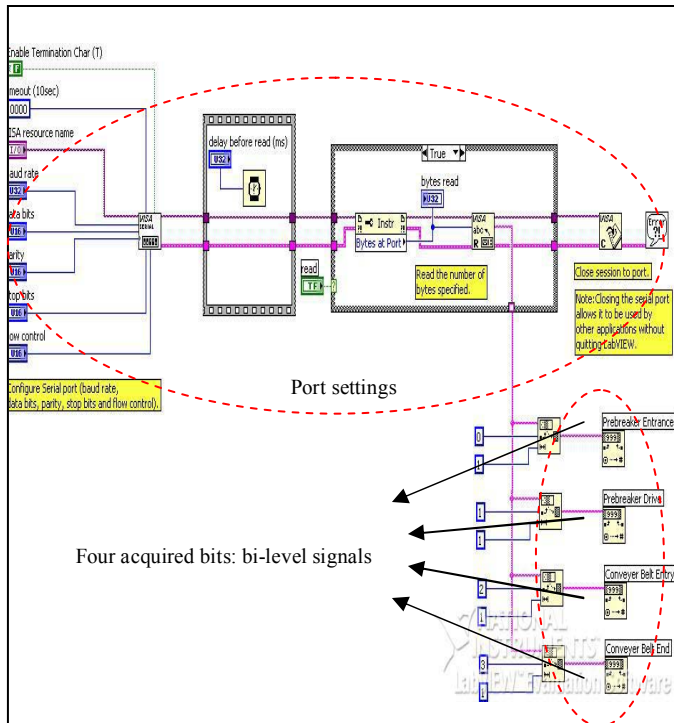


Figure 5. Partial Schematic of Data Fusion Unit

TABLE I
SELECTION OF SENSORS FOR THE DESCRIBED MANUFACTURING PROCESS

| Site | Sensor Type | Data Type |
|-----------------------|------------------|-------------|
| Prebreaker Entrance | Light Reflection | Bi-level |
| Prebreaker Drive | Current Sensor | Multi-level |
| Conveyer Belt 1 Entry | Light Reflection | Bi-level |
| Conveyer Belt 1 End | Light Reflection | Bi-level |
| Grinder Motor Drive | Current Sensor | Bi-level |
| Conveyer Belt 2 Entry | Light Reflection | Bi-level |
| Conveyer Belt 2 End | Light Reflection | Bi-level |
| Silo Top | Range Detector | Multi-level |
| Conveyer Belt 3 | Speed Detector | Multi-level |
| Mixer Motor Drive | Current Sensor | Bi-level |
| Conveyer Belt 4 | Speed Detector | Multi-level |
| Meat Pump Drive | Current Sensor | Bi-level |
| Conveyer Belt 5 | Speed Detector | Multi-level |
| Mixer Motor Drive | Current Sensor | Bi-level |
| Conveyer Belt 6 | Speed Detector | Multi-level |
| Siefer Motor Drive | Current Sensor | Bi-level |
| Conveyer Belt 7 | Speed Detector | Multi-level |
| Emulsion Motor Drive | Current Sensor | Bi-level |
| Tunnel | Speed Detector | Multi-level |
| Tunnel | Thermometer | Multi-level |
| Conveyer Belt 8 | Speed Detector | Multi-level |
| Filler | Light Reflection | Bi-level |
| Filler | Range Detector | Multi-level |
| Seamer | Light Reflection | Bi-level |
| Conveyer Belt 9 Entry | Light Reflection | Bi-level |
| Conveyer Belt 9 End | Light Reflection | Bi-level |
| FMC | Switch | Bi-level |
| Stork | Switch | Bi-level |
| Labeller | Light Reflection | Bi-level |
| Kister | Switch | Bi-level |
| Palletiser | Switch | Bi-level |
| Wrapper | Switch | Bi-level |

Conveyer Belt 1 Entry, Conveyer Belt 2 End, and Grinder Motor Drive signals. The rest of the signals are acquired with similar Virtual Instrument design.

G. DFU Development

The signals obtained from the DIU, is combined together in order to extract the required information. Glue logic depends on the type of raw acquired data and also needed information. For example, time factors may be deduced from difference of times measured between two triggering signals of a bi-level signal. Alternatively, another time factor may be the time interval measured between two separate and sequential bi-level sensors on the conveyer belt. Figure 6 demonstrates how two signals of different type may build up a new type of information. Pulse signals received from the bit representing Prebreaker Entrance sensor output may seem enough for building up meat arrival period information. However, for redundancy purposes, one may measure the level of current sought by the Prebreaker's motor drive circuit. If it is above a certain threshold it could also indicate that a piece of meat has

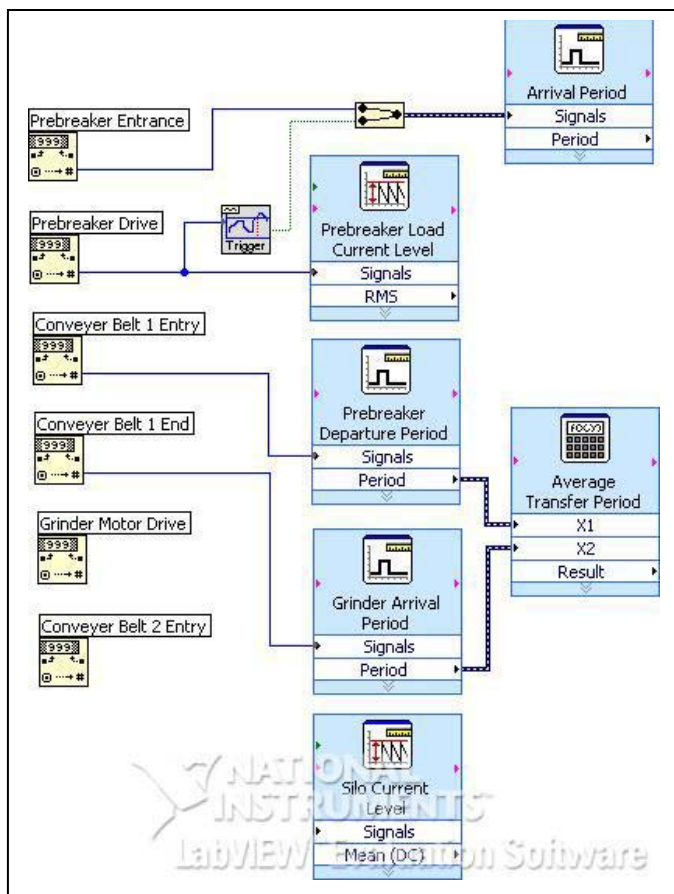


Figure 6. Partial Schematic of Data Fusion Unit

entered the machine. Therefore, there can be a situation where Prebreaker Entrance pulses and high motor current together ensure entry of a product into the process. Periods between such entries may be measured and passed for further analysis. Similarly, by averaging between two arrival and departure periods of product on the conveyer belt 1, the average period of product transfer is obtainable.

In building DFU, Waveform Measurement, Mathematics and Signal Processing virtual instruments help more than the others. However, a decent variety of virtual instruments is available to the designer of the DFU.

IV. CONCLUSION

In this paper the importance of a comprehensive environment for design and development of data interface units and data fusion units was emphasized. The shortcomings of the existing data acquisition and integration systems in terms of data input management were highlighted. Flexible Data Input Layer Architecture (FDILA) was proposed to fill the existing gap in current shop floor data management system. The proposed architecture enables the system

developers (line managers) to define any number of data entry points and data types in a data integration system no longer needs proprietary firmware system.

The Labview implementation environment introduced here has more of a technical and application programming character than a top layer configuration user interface. Extensions to this work is currently being made to connect the fused information to higher level information management system where shop floor Key Performance Factors (KPF) are measured for improved quick response decision making.

ACKNOWLEDGMENT

The authors acknowledge the financial and continuous support provided by the Engineering and Physical Sciences Research Council (EPSRC) of the United Kingdom.

REFERENCES

- [1] S. S. Iyengar, D. N. Jayasimha and D. Nadig, "A versatile architecture for the distributed sensor integration problem," *IEEE Trans. Computers*, vol. 43, no. 2, Feb. 1994, pp. 175-185.
- [2] I. K. Ibrahim, R. Kronsteiner, and G. Kotsis, "A semantic solution for data integration in mixed sensor networks," *Computer Communications*, vol. 28, pp. 1564-1574, Apr. 2005.
- [3] A. Mousavi, M. Hamdi, M. Sarhadi, "Knowledge-Based quick response decision making in food processing industry," *International Journal of Systems Engineering*, vol. 2, no. 1, pp. 99-122, 2007.
- [4] M. Clayton, J. Durham, S. Felker, G. Fortner, J. Martinez and S. Shelton, "Manufacturing integration of data collection and statistical process control systems," in *Proc. IEEE 6th International Conf. Emerging Technologies and Factory Automation*, Los Angeles, CA, 1997, pp. 505-507.
- [5] S. Tavakoli, A. Mousavi, "Input Variable Monitoring for Quick-Response Decision Making in Volatile Systems," *Proceedings of Virtual Concept 2006*, Playa Del Carmen, Mexico, 2006.
- [6] F. Cecelja, *Manufacturing information and data systems*. Great Britain: Penton Press, 2002, ch. 4.
- [7] T. Perera, K. Liyanage, "Methodology for rapid identification and collection of input data in the simulation of manufacturing systems," *Simulation Practice and Theory*, vol. 7, pp. 645-656, 2000.
- [8] H. Eren, *Wireless Sensors and Instruments: Networks, Design, and Applications*, Boca Raton, FL: Taylor and Francis, 2006, pp. 29-32.
- [9] R. L. Lonsbury, "Remote parameter set up and production data collection on armature manufacturing work cells," Unpublished.
- [10] M. J. Moreno-Lizaranzu, R. A. Wysk, J. Hong and V. V. Prabhhu, "A hybrid shop-floor control system for food manufacturing," *IIE Trans.* Vol. 33, no. 3, pp. 193-202, Mar. 2001.
- [11] C. Hsu, "Manufacturing information systems," in *Handbook of Design, Manufacturing, and Automation*, R. C. Dorf and A. Kusiak, Eds. Canada: John Wiley and Sons, 1994, pp. 737-766.
- [12] S. Ozdemir, "Concealed data aggregation in heterogeneous sensor networks using privacy homomorphism," in *Proc. IEEE Int. Conf. Pervasive Services*, Istanbul, Turkey, Jul. 2007.
- [13] B. Zhou, L. H. Ngoh, B. S. Lee, and C. P. Fu, "A hierarchical scheme for data aggregation in sensor network in Conf. Rec. 2004 12th IEEE Int. Conf. Networks, vol. 2, pp. 525-529.
- [14] B. Thuraisingham, "Dependable infrastructures and data managers for sensor networks," in *Proc. 9th IEEE Int. Workshop. Object-Oriented Real-Time Dependable Systems*, pp. 11-16, Oct. 2003.
- [15] R. Mueller, G. Alonso, and D. Kossmann, "SwissQM: Next generation data processing in sensor networks," in *Proc. 3rd Conf. Innovative Data Systems Research*, Asilomar, California: Jan. 2007.